

Minimum Falling Path Sum [\(View\)](#)

Given an $n \times n$ array of integers matrix, return the **minimum sum** of any **falling path** through matrix.

A **falling path** starts at any element in the first row and chooses the element in the next row that is either directly below or diagonally left/right. Specifically, the next element from position (row, col) will be (row + 1, col - 1), (row + 1, col), or (row + 1, col + 1).

Example 1:

2	1	3
6	5	4
7	8	9

2	1	3
6	5	4
7	8	9

2	1	3
6	5	4
7	8	9

Input: matrix = [[2,1,3],[6,5,4],[7,8,9]]

Output: 13

Explanation: There are two falling paths with a minimum sum as shown.

Example 2:

-19	57
-40	-5

-19	57
-40	-5

Input: matrix = [[-19,57],[-40,-5]]

Output: -59

Explanation: The falling path with a minimum sum is shown.

Constraints:

- $n == \text{matrix.length} == \text{matrix}[i].\text{length}$
- $1 \leq n \leq 100$
- $-100 \leq \text{matrix}[i][j] \leq 100$